

LojaInteligente

Documentação

Sumário

[Aplicações](#)

[Loja Virtual \(front-end\)](#)

[Tecnologias](#)

[Desenvolvimento](#)

[Servidor](#)

[Estrutura](#)

[app/](#)

[deploy/](#)

[dist/](#)

[lojainteligente_config/](#)

[Desenvolvimento](#)

[Instalando Node.js e NPM](#)

[Rodando a aplicação a primeira vez](#)

[Rodando a aplicação em modo desenvolvimento](#)

[Encerrando a execução do modo desenvolvimento](#)

[Implementação \(Deploy\)](#)

[Ambiente de Homologação](#)

[Ambiente de Produção](#)

[API \(backend\)](#)

[Tecnologias](#)

[Desenvolvimento](#)

[Implementação](#)

[Estrutura](#)

[.meteor/](#)

[deploy/](#)

[model/](#)

[packages/](#)

[Desenvolvimento](#)

[Instalando Meteor.js](#)

[Rodando a aplicação a primeira vez](#)

[Rodando a aplicação em modo desenvolvimento](#)

[Encerrando a execução do modo desenvolvimento](#)

[Implementação \(Deploy\)](#)

[Ambiente de Homologação](#)

[Ambiente de Produção](#)

[Backoffice](#)

[Tecnologias](#)

[Desenvolvimento](#)

[Implementação](#)

[.meteor/](#)

[client/](#)

[deploy/](#)

[model/](#)

[packages/](#)

[public/](#)

[server/](#)

[Desenvolvimento](#)

[Instalando Meteor.js](#)

[Rodando a aplicação a primeira vez](#)

[Rodando a aplicação em modo desenvolvimento](#)

[Encerrando a execução do modo desenvolvimento](#)

[Implementação \(Deploy\)](#)

[Ambiente de Homologação](#)

[Ambiente de Produção](#)

Serviços de Terceiros

[DigitalOcean](#)

[Cloudflare](#)

[IBM Cloud - MongoDB](#)

[AWS - S3](#)

[YaPay](#)

[Sparkpost](#)

[Prerender](#)

Dúvidas

Aplicações

Loja Virtual (front-end)

Tecnologias

Desenvolvimento

- AngularJs (Framework JavaScript para front-end) - [documentação](#)
- Webpack (module bundler JavaScript) - [documentação](#)
- HTML5
- CSS3

Servidor

- Nginx

Estrutura

app/

Pasta onde ficam todos os arquivos de desenvolvimento. Para maiores informações, consulte a [documentação do AngularJs](#).

deploy/

Pasta com os scripts de deploy para ambientes de produção e homologação.

dist/

Pasta onde a aplicação é *buildada* antes do de realizar o deploy.

lojainteligente_config/

Pasta de configurações de conexão com a API de acordo com o ambiente.

Desenvolvimento

Instalando Node.js e NPM

Para trabalhar em ambiente de desenvolvimento você precisará ter Node.js e NPM instalado.

Basta baixar e instalar conforme este link: <https://www.npmjs.com/get-npm>

Rodando a aplicação a primeira vez

Para rodar a aplicação a primeira vez, será necessário instalar todas suas dependências.

Acesse a pasta do projeto da aplicação através do terminal do seu computador e execute o comando:

```
npm install
```

Após terminar a instalação o projeto estará pronto para ser rodado em modo desenvolvimento.

Rodando a aplicação em modo desenvolvimento

Na pasta do projeto da aplicação execute o comando:

Utilizando a API do ambiente de homologação:

```
npm start
```

Utilizando a API rodando localmente:

Atenção! Para isso você precisará rodar a API em sua máquina seguindo os [passos em API > Desenvolvimento presentes nesta documentação](#).

```
npm run local
```

Mantenha o terminal aberto durante o desenvolvimento.

Você pode acessar a aplicação acessando <http://localhost:8080> em seu navegador.

Toda vez que você alterar e salvar um arquivo, ou adicionar um novo arquivo em qualquer subpasta da pasta **app**, a aplicação será atualizada automaticamente no navegador.

Encerrando a execução do modo desenvolvimento

Para encerrar a execução da aplicação, basta acessar o terminal onde ela está sendo executada e pressionar:

- **Ctrl + C** para Windows e Linux
- **Cmd + C** para MacOS

Implementação (Deploy)

Ambiente de Homologação

Para implementar uma nova versão em ambiente de homologação, acesse a pasta do projeto da aplicação através do terminal de seu computador e execute o comando:

```
npm run staging
```

O *Webpack* irá gerar o *bundle* e em seguida você será solicitado a digitar a senha do servidor. Digite e pressione **ENTER**.

As alterações serão enviada para o servidor.

Para visualizar pode ser preciso limpar o cache no serviço [Cloudflare](#).

Logo após, acesse o endereço de homologação da sua aplicação.

Ambiente de Produção

Para implementar uma nova versão em ambiente de produção, acesse a pasta do projeto da aplicação através do terminal de seu computador e execute o comando:

```
npm run production
```

O *Webpack* irá gerar o *bundle* e em seguida você será solicitado a digitar a senha do servidor. Digite e pressione **ENTER**.

As alterações serão enviada para o servidor.

Para visualizar pode ser preciso limpar o cache no serviço [Cloudflare](#).

Logo após, acesse o endereço de produção da sua aplicação.

API (backend)

Tecnologias

Desenvolvimento

- MeteorJs (Framework backend) - [documentação](#)
- NodeJS (Servidor backend) - [site](#)
- MongoDB (Banco de Dados) - [documentação](#)

Implementação

- MeteorUp (mup) - [documentação](#)

Estrutura

.meteor/

Pasta utilizada pelo framework MeteorJs para os arquivos necessários durante o desenvolvimento. Não é recomendado alterar nenhum arquivo nela.

deploy/

Pasta com os scripts de deploy para ambientes de produção e homologação.

model/

Pasta contendo os arquivos das *Collections* que fazem a comunicação com as coleções do banco de dados. Para maiores informações, consulte a [documentação sobre Collections do MeteorJs](#).

packages/

Pasta contendo os arquivos de desenvolvimento dos pacotes.

Cada módulo da API está separado em um pacote. Para maiores informações, consulte a [documentação sobre Packages do MeteorJs](#) e a [documentação do MeteorJs](#).

A API utiliza o pacote **mrest:restivus**. Para maiores informações, consulte a [documentação do pacote Restivus](#).

Desenvolvimento

Instalando Meteor.js

Para trabalhar em ambiente de desenvolvimento você precisará instalar o MeteorJs, siga os passos neste link: <https://www.meteor.com/install>.

Rodando a aplicação a primeira vez

Para rodar a aplicação a primeira vez, será necessário instalar todas suas dependências.

Acesse a pasta do projeto da aplicação através do terminal do seu computador e execute o comando:

```
npm install
```

Após terminar a instalação o projeto estará pronto para ser rodado em modo desenvolvimento.

Rodando a aplicação em modo desenvolvimento

Na pasta do projeto da aplicação execute o comando:

```
npm start
```

Atenção! A primeira execução pode levar vários minutos ou mesmo horas para baixar e instalar todas as dependências do MeteorJs. Não é aconselhável parar o processo sem ter concluído.

Mantenha o terminal aberto durante o desenvolvimento.

Você pode acessar a aplicação acessando <http://localhost:3010> usando um Client REST como o [Postman](#) ou [Insomnia](#).

Encerrando a execução do modo desenvolvimento

Para encerrar a execução da aplicação, basta acessar o terminal onde ela está sendo executada e pressionar:

- **Ctrl + C** para Windows e Linux
- **Cmd + C** para MacOS

Implementação (Deploy)

Ambiente de Homologação

Para implementar uma nova versão em ambiente de homologação, acesse a pasta do projeto da aplicação através do terminal de seu computador e execute o comando:

```
npm run staging
```

O *MeteorUp* irá gerar o *bundle*, subir a aplicação para o servidor, executar e testar a execução. Este processo pode levar alguns minutos.

Logo após, acesse o endereço de homologação da sua aplicação.

Ambiente de Produção

Para implementar uma nova versão em ambiente de produção, acesse a pasta do projeto da aplicação através do terminal de seu computador e execute o comando:

```
npm run production
```


O *MeteorUp* irá gerar o *bundle*, subir a aplicação para o servidor, executar e testar a execução. Este processo pode levar alguns minutos.

Logo após, acesse o endereço de produção da sua aplicação.

Backoffice

Tecnologias

Desenvolvimento

- AngularJs (Framework JavaScript para front-end) - [documentação](#)
- MeteorJs (Framework backend) - [documentação](#)
- NodeJS (Servidor backend) - [site](#)
- MongoDB (Banco de Dados) - [documentação](#)
- Fuse AngularJs (Template) - [documentação](#)
- HTML5
- CSS3

Implementação

- MeteorUp (mup) - [documentação](#)

`.meteor/`

Pasta utilizada pelo framework MeteorJs para os arquivos necessários durante o desenvolvimento. Não é recomendado alterar nenhum arquivo nela.

`client/`

Pasta com o front-end da aplicação. Para mais informações, consulte as documentações:

- [AngularJs](#)
- [Fuse AngularJs](#)
- [MeteorJs](#)

`deploy/`

Pasta com os scripts de deploy para ambientes de produção e homologação.

`model/`

Pasta contendo os arquivos das *Collections* que fazem a comunicação com as coleções do banco de dados. Para maiores informações, consulte a [documentação sobre Collections do MeteorJs](#).

packages/

Pasta contendo os arquivos de desenvolvimento dos pacotes.

Todos os módulos do Fuse AngularJs estão em formato de pacotes, assim como alguns módulos utilizados pelo AngularJs.

Para maiores informações, consulte a [documentação sobre Packages do MeteorJs](#) e a [documentação do MeteorJs](#).

public/

Pasta contendo os ativos públicos da aplicação, como por exemplo imagens e ícones.

server/

Pasta contendo os *scripts server-side* da aplicação. Para maiores informações, consulte a [documentação do MeteorJs](#).

Desenvolvimento

Instalando Meteor.js

Para trabalhar em ambiente de desenvolvimento você precisará instalar o MeteorJs, siga os passos neste link: <https://www.meteor.com/install>.

Rodando a aplicação a primeira vez

Para rodar a aplicação a primeira vez, será necessário instalar todas suas dependências.

Acesse a pasta do projeto da aplicação através do terminal do seu computador e execute o comando:

```
npm install
```

Após terminar a instalação o projeto estará pronto para ser rodado em modo desenvolvimento.

Rodando a aplicação em modo desenvolvimento

Na pasta do projeto da aplicação execute o comando:

```
npm start
```

Atenção! A primeira execução pode levar vários minutos ou mesmo horas para baixar e instalar todas as dependências do MeteorJs. Não é aconselhável parar o processo sem ter concluído.

Mantenha o terminal aberto durante o desenvolvimento.

Você pode acessar a aplicação acessando <http://localhost:3000> em seu navegador.

Encerrando a execução do modo desenvolvimento

Para encerrar a execução da aplicação, basta acessar o terminal onde ela está sendo executada e pressionar:

- **Ctrl + C** para Windows e Linux
- **Cmd + C** para MacOs

Implementação (Deploy)

Ambiente de Homologação

Para implementar uma nova versão em ambiente de homologação, acesse a pasta do projeto da aplicação através do terminal de seu computador e execute o comando:

```
npm run staging
```

O *MeteorUp* irá gerar o *bundle*, subir a aplicação para o servidor, executar e testar a execução. Este processo pode levar alguns minutos.

Logo após, acesse o endereço de homologação da sua aplicação.

Ambiente de Produção

Para implementar uma nova versão em ambiente de produção, acesse a pasta do projeto da aplicação através do terminal de seu computador e execute o comando:

```
npm run production
```

O *MeteorUp* irá gerar o *bundle*, subir a aplicação para o servidor, executar e testar a execução. Este processo pode levar alguns minutos.

Logo após, acesse o endereço de produção da sua aplicação.

Serviços de Terceiros

DigitalOcean

Servidores virtuais onde ficam rodando as aplicações.

Acesso: <https://cloud.digitalocean.com/login>

Cloudflare

Serviço de DNS e CDN para distribuição e cache do conteúdo estático (imagens e código).

Acesso: <https://dash.cloudflare.com/login>

IBM Cloud - MongoDB

Serviço de banco de dados.

Acesso: <https://cloud.ibm.com/login>

AWS - S3

Serviço de armazenamento das imagens.

Acesso: <https://console.aws.amazon.com>

YaPay

Serviço de gateway de pagamento - faz a interface entre as aplicações e os meios de pagamentos (adquirentes de cartão de crédito, bancos para emissão de boleto e outros).

Acesso: <https://gateway.yapay.com.br/administrador/>

Sparkpost

Serviço de envio dos e-mails transacionais (recuperação de senha, status do pedido, etc) das aplicações.

Acesso: <https://app.sparkpost.com/auth>

Prerender

Serviço de cache do conteúdo dinâmico da Loja (aplicação front-end) para permitir que robôs de mecanismos de busca (como o Google) indexem o conteúdo da loja virtual.

Acesso: <https://prerender.io/login>

Dúvidas

Em caso de dúvidas entre em contato:

Guima Ferreira

WhatsApp: (31) 99697-54098